

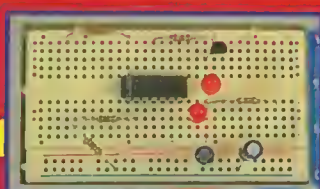
impara elettronica digitale

...e costruisci il tuo **LABORATORIO DIGITALE**

6,90 €



HARDWARE



DIGITALE DI BASE

```
FOR x=0 to 10 DO
...
END

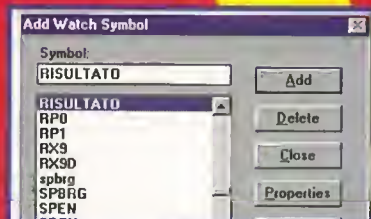
x=0
DO
  x=x+1
UNTIL x=10

x=0
WHILE x<10 DO
  x=x+1
END

x=0
DO
  x=x+1
WHILE x<10
```

MICROCONTROLLER

18



DIGITALE AVANZATO

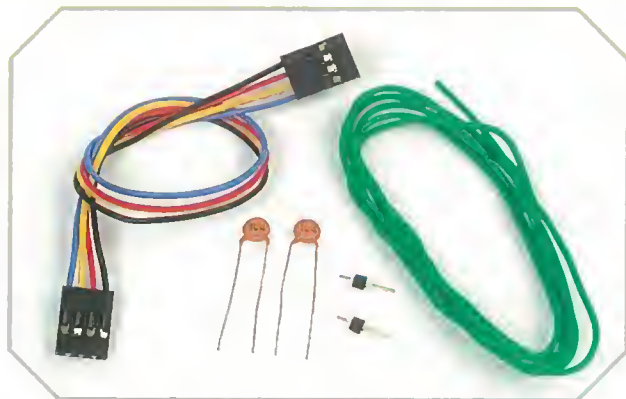


Peruzzo & C.

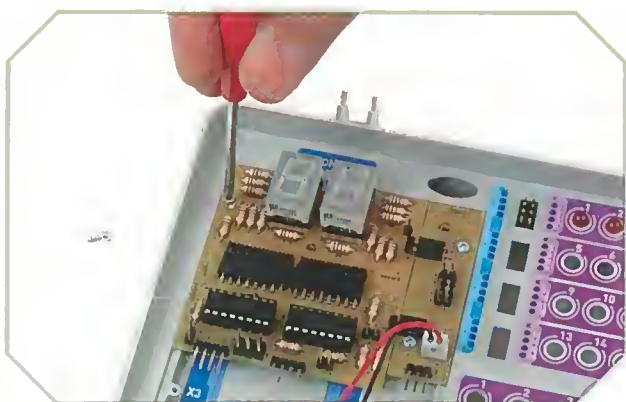
**TOTALMENTE
PROGRAMMABILE!!!**



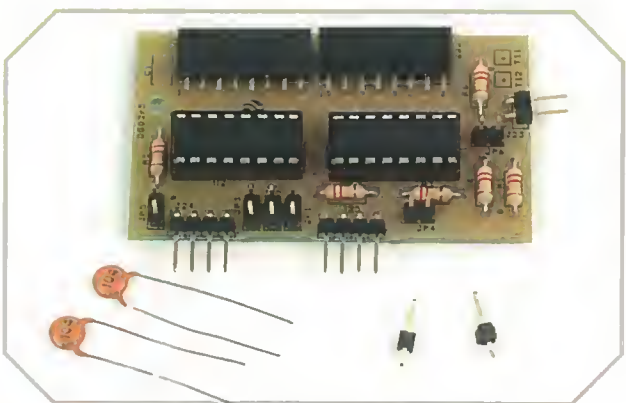
Cavi e accessori



Materiali allegati a questo fascicolo.



Togliamo le viti delle schede DG01 e DG02, e allentiamo quelle di DG04.



La scheda DG02 e i componenti in attesa di essere installati sulla scheda stessa.

Con questo fascicolo viene fornito un altro cavetto di collegamento, terminato su una coppia di connettori a quattro terminali, oltre a del filo rigido per aumentare le possibilità di collegamento tra i terminali della Bread Board e da questa verso l'esterno. A questo si aggiungono i due connettori e i due condensatori di filtro dell'alimentazione per completare la scheda del driver DG02.

Scheda DG02

Il primo lavoro a cui ci dedicheremo sarà completare il circuito stampato DG02, smontandolo faremo attenzione a scollegare simultaneamente DG01 e DG02 da DG04. È necessario togliere completamente le viti che fissano le schede DG01 e DG02 e allentare di un giro le due viti che fissano la scheda DG04.

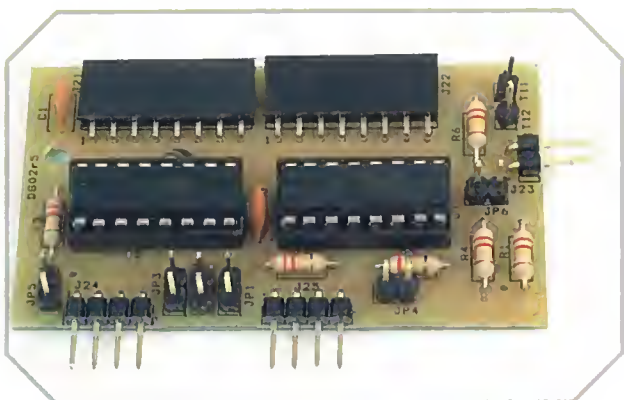
Dopo aver tolto la scheda si inseriscono i terminali nei fori indicati sulla serigrafia della scheda come T11 e T12, si volta la scheda e si salda dal lato apposito per la saldatura.

I terminali dei due condensatori di filtro dell'alimentazione, che sono del tipo ceramico da 100 nF di capacità, si inseriscono nei fori indicati sulla scheda come C1 e C2, in seguito si salderanno e si taglierà la parte in eccesso dei reofori di entrambi.

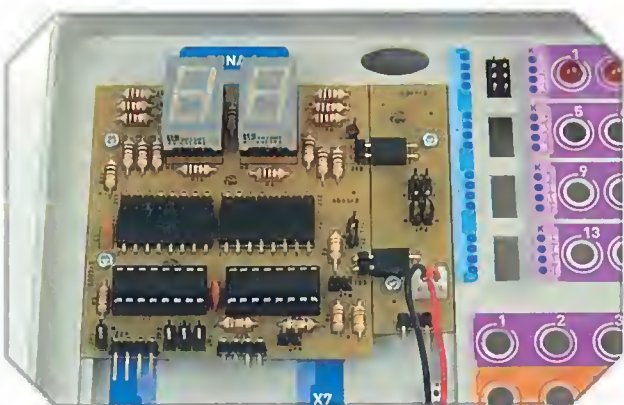
Con queste operazioni la scheda DG02 è totalmente terminata. Questa scheda si assembla con la DG01 ed entrambe si collegano alla DG04. Si montano le viti di fissaggio delle schede DG01 e DG02, dopodiché si stringono le viti della DG04. Queste viti devono essere chiuse stringendole quanto basta per fissare le schede, non devono essere strette più del necessario in quanto si potrebbero danneggiare i filetti che le viti stesse creano nelle torrette di fissaggio delle schede.



C1 e C2 sono i condensatori di filtro dell'alimentazione.



Scheda DG02 completa.



Reinstallazione dei circuiti stampati DG01, DG02 e DG04.

Collegamenti con la Bread Board

Vi è stato fornito un filo verde lungo circa 1 m, lo taglieremo in parti più corte per adattarlo ai collegamenti tra i vari punti della Bread Board e tra questa e gli altri elementi del laboratorio, in special modo con le molle di collegamento che vi verranno fornite prossimamente. Non c'è una regola fissa per fare questi pezzi, dato che sulla Bread Board si può lavorare in molti modi.

Il nostro consiglio è di dividere questo filo e quello fornito in precedenza per ottenere un insieme di fili di collegamento, spelati agli estremi per circa 6 mm, con le lunghezze indicate di seguito:

12 da 5 cm.

2 da 10 cm.

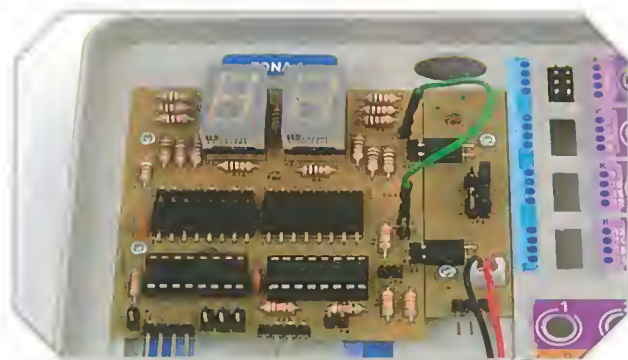
8 da 15 cm.

Queste misure sono una proposta, non è necessario che siano molto precise, si possono tagliare i fili anche in altro modo, l'importante è che sia possibile realizzare i collegamenti proposti; potete inoltre utilizzare qualsiasi altro filo che non superi 0,5 mm di diametro. Bisogna fare attenzione a non danneggiare il conduttore di rame togliendo l'isolante dalle estremità, per fare in modo che non si rompa.

Portabatterie

Con il fascicolo precedente vi sono state fornite quattro viti, due delle quali sono già state utilizzate, le altre due le utilizzeremo ora per montare il secondo portabatterie.

Per installare questo portabatterie gireremo il laboratorio e lo presenteremo nella sua sede, facendo in modo che i suoi fori rimangano allineati con quelli delle torrette del laboratorio, quindi inseriremo una vite e la avviteremo fino in fondo senza stringere, per poter centrare anche l'altro foro; eseguita questa operazione, inseriremo anche l'altra vite e termineremo il fissaggio del portabatterie chiudendo a fondo entrambe le viti. In ogni caso ricordate di non stringere in eccesso.



Prova col filo con due terminali femmina tra il terminale di uscita dell'alimentazione della scheda T1 e uno di quelli appena installati T11 e T12. Il punto di uno dei display si deve accendere.



Il secondo portabatterie si installa utilizzando due viti.

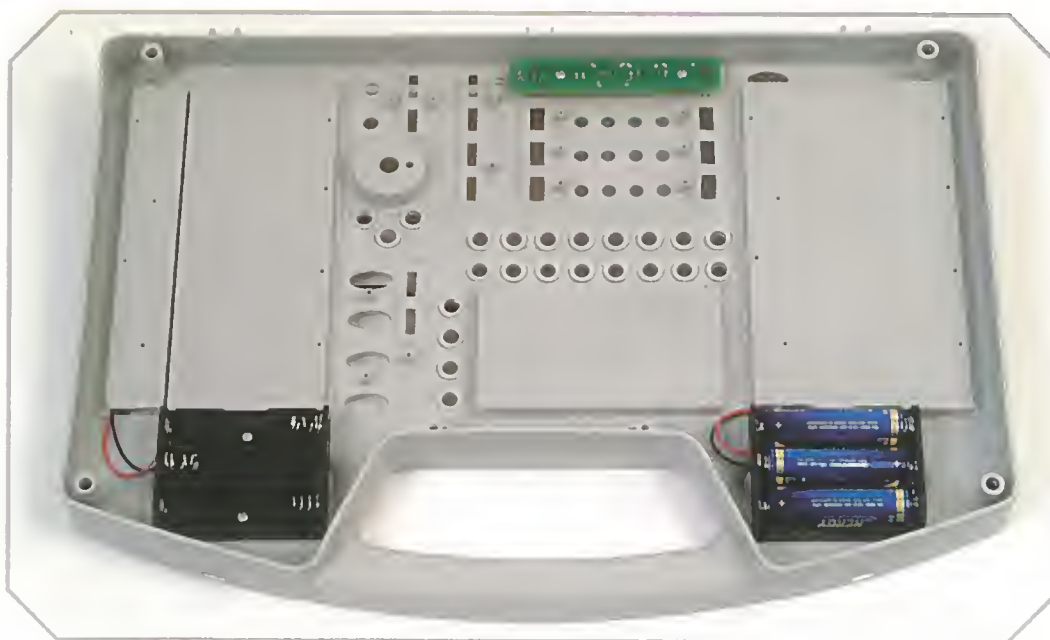
Prova dei terminali T11 e T12

I terminali T11 e T12 della scheda DG02 corrispondono al collegamento della resistenza di polarizzazione dell'anodo del LED, che rappresenta il punto situato nella parte inferiore di ogni display a sette segmenti. Se osserviamo il circuito stampato potremo contare 16 resistenze, una per ogni segmento e due per i due punti. Il driver non è progettato per controllare l'accensione di questi due LED, quindi sarà necessario utilizzare un collegamento esterno quando vorremo farli illuminare.

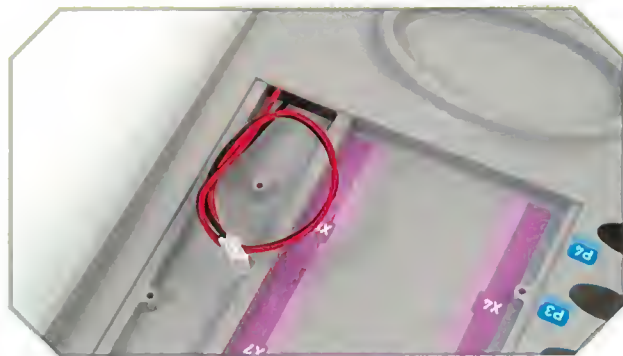
L'illuminazione di questi punti si ottiene applicando una tensione positiva ai terminali T11 e T12 della scheda DG02. Questa tensione positiva deve essere compresa tra 4,5 e 9 volt e si può ottenere dall'uscita ausiliaria T1 della scheda DG01.

Per realizzare questa prova bisogna montare tre pile da 1,5 volt nel portabatterie situato sotto la zona 1, inoltre bisogna inserire i ponticelli tra 1-2 e 1-2 della scheda DG04.

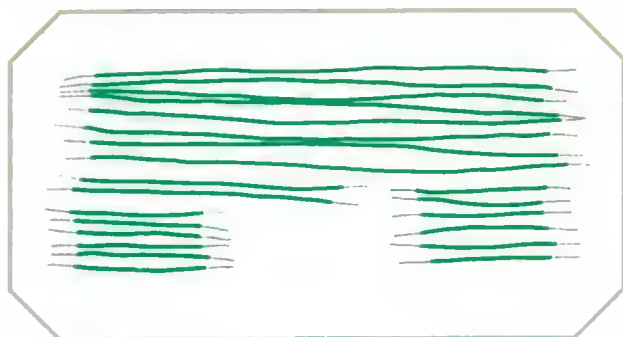
Se colleghiamo un estremo del filo con connettore femmina sul terminale T1 e l'altro estremo a T11, si illumina il punto di uno dei display, se ci spostiamo su T12 si illumina il punto dell'altro display.



Aspetto del pannello principale del laboratorio visto da sotto.



Per il momento il cavo del portabatterie si deve arrotolare nella zona 2 del laboratorio.



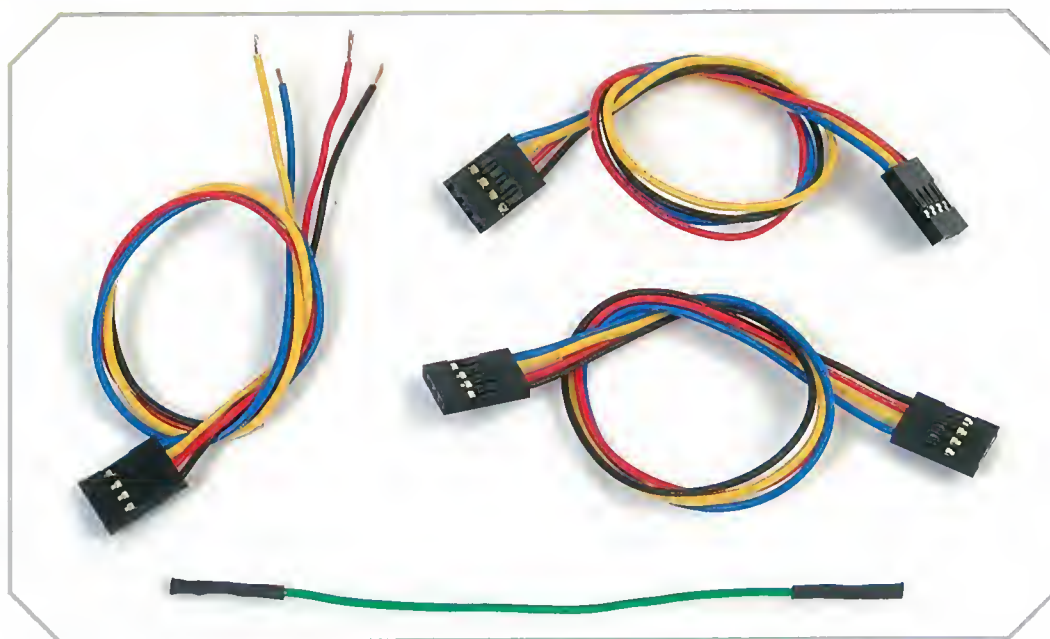
Insieme di fili di connessione.

I cavetti

I cavetti di collegamento con connettori a quattro terminali hanno diverse applicazioni, dato che i collegamenti dei circuiti stampati sono raggruppati, per quanto possibile, di 4 in 4. Anche la scheda dei pulsanti, che non vi è ancora stata fornita, e i collegamenti dei LED, son raggruppati di 4 in 4.

Per quanto riguarda l'alimentazione, li utilizzeremo provvisoriamente in questi fascicoli, dato che collegando tutte le schede l'alimentazione si trasferirà da una all'altra, inoltre le molle indicate come 0 V, 5 V, 9 V e V saranno alimentate dalla parte inferiore del pannello principale, quindi i collegamenti tra la Bread Board e queste molle di connessione si potranno realizzare direttamente con uno o due pezzi di filo con gli estremi spelati. Fino a questo momento vi abbiamo fornito solamente due cavi terminati su due connettori a quattro terminali, uno terminato su un connettore e un cavetto ausiliario che ha un solo filo terminato su un connettore femmina a ogni estremo.

Ve ne forniremo altri con la lunghezza e il numero di terminali necessari per realizzare tutti gli esperimenti proposti, dato che, andando avanti, questi si complicheranno sempre di più, avremo a disposizione più materiali e vi verranno spiegati più argomenti.



Questi sono i cavetti di collegamento già forniti.

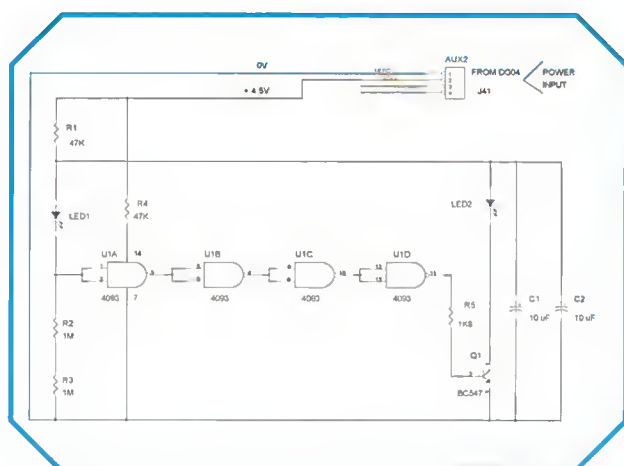


LED a basso consumo

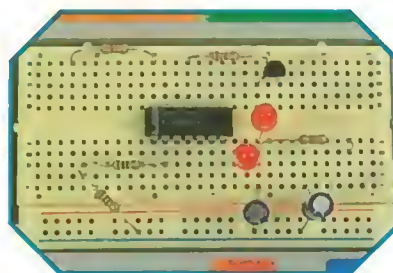
Questo circuito è un esempio di come, complicando un po' l'elettronica, è possibile ottenere un indicatore luminoso a LED in più a basso consumo. Un LED costantemente acceso ha bisogno di almeno 3 mA per potersi illuminare, consumo molto elevato per un dispositivo alimentato a batterie. Con questo circuito invece si dispone di un LED che emette un lampeggio all'incirca ogni 3 o 5 secondi, con un consumo inferiore a 40 mA, quindi può funzionare molti giorni senza cambiare le batterie.

Lo schema

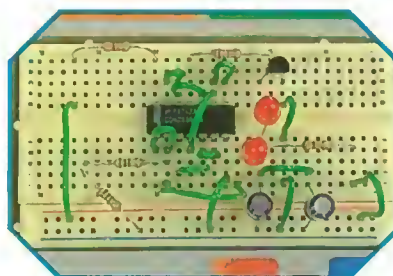
Il circuito è semplice, però un po' particolare. Si tratta fondamentalmente di un circuito di controllo che funziona solamente quando il condensatore C1 + C2 si carica tramite la resistenza R1 da 47 K e inoltre quando all'ingresso del circuito — terminali 1 e 2 dell'integrato — si raggiunge il livello logico 1. Quando questo ingresso passa a livello 1 l'uscita del circuito, terminale 11 del circuito integrato, passa anch'essa a livello 1, portando in conduzione il transistor Q1, quindi il LED 2 si illumina. In questo momento viene consumata in modo quasi istantaneo l'energia accumulata nel condensatore formato dalla somma di C1 e C2. Nello stesso istante, come conseguenza del fatto che il condensatore si è scaricato, la tensione sull'ingresso di controllo si abbassa: il circuito interpreta questo come un livello basso, quindi il LED cessa di condurre. Il LED 1 si utilizza per mantenere una differenza di potenziale e il suo consumo è così ridotto che non si illumina nemmeno.



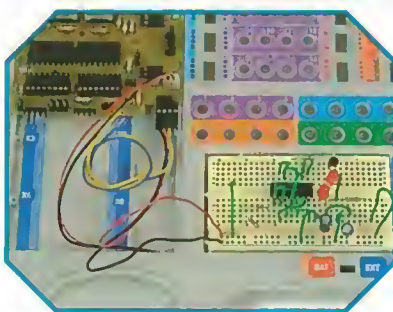
Schema elettrico del LED a basso consumo.



Componenti inseriti sulla Bread Board.



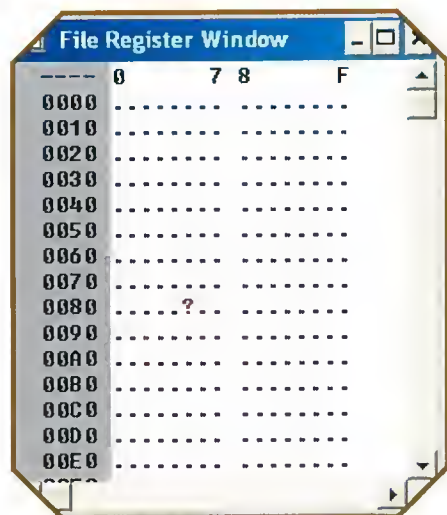
Cablaggio dei collegamenti.



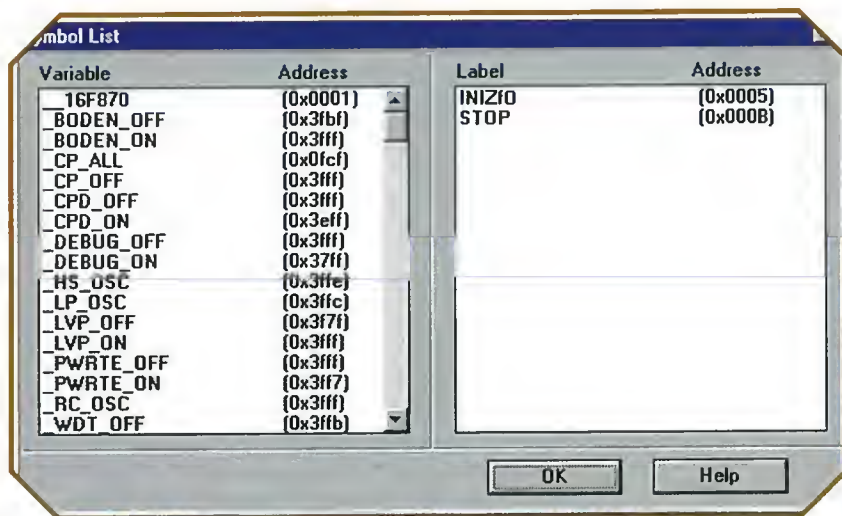
Alimentazione tramite J41 di DG04, utilizzando questa volta il cavetto di un connettore.

L'attivazione

Quando si collega l'alimentazione del circuito, gli ingressi della porta U1A — terminali 1 e 2 del circuito integrato — sono a livello basso, e le resistenze R2 e R3 li mantengono a livello basso. Questa situazione dura piuttosto a lungo, perché la carica del condensatore è lenta, in quanto si produce tramite una resistenza da 47 K. Tramite il LED 1 non circola corrente, quindi non c'è attivazione fino a quando non viene superata la sua soglia di conduzione, do-



I registri di utilizzo generale in modo ASCII.



Listato dei simboli utilizzati nel programma.

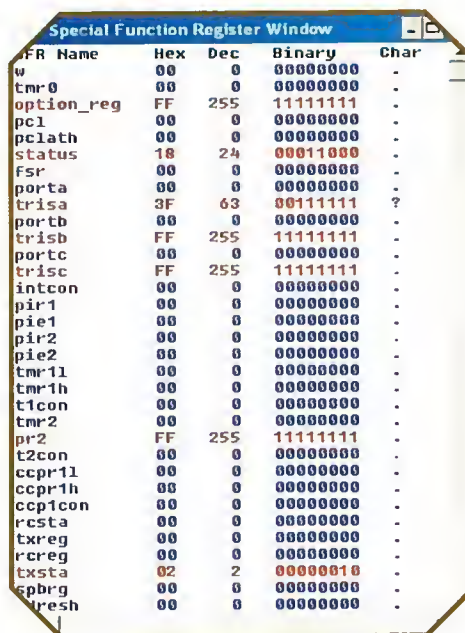
Special Function Registers

Selezionando Window → Special Function Registers potremo vedere il contenuto dei registri con funzioni speciali (FSR). Durante la simulazione questa è una delle finestre da tenere normalmente aperta, il formato fornito da questa finestra risulta molto utile per analizzare il contenuto di questi registri in ogni momento. Come mostra la figura, in questa finestra appaiono tutti i registri FSRs identificati con il loro nome e con il loro contenuto nei formati esadecimale, decimale, binario e ASCII.

Per modificare il contenuto di uno di questi registri dobbiamo fare doppio clic sul registro prescelto e immediatamente apparirà la finestra Modify, che permetterà la scrittura del registro.

Show Symbol List

In questa finestra sono riportati i simboli delle variabili e delle etichette, utilizzati nel codice sorgente del nostro programma. Dopo aver compilato il programma si genera un file con estensione ".cod" che contiene la definizione di tutti i simboli utilizzati nel programma. Questa finestra ha una funzione informativa riguardo ai simboli che abbiamo utilizzato e serve unicamente come orientamento in modo da poter identificare gli indirizzi di memoria a cui sono associati questi simboli.



Finestra dei registri di funzione speciali (FSRs).

Stopwatch

Durante la simulazione spesso ci tornerà utile poter calcolare il tempo di esecuzione di un programma o di una subroutine. Possiamo fare questo contando il numero di istruzioni che si realizzano e moltiplicandolo per quattro volte la frequenza del segnale di clock, o per otto nel caso in cui le istruzioni siano di salto, oppure lo potremo fare utilizzando il nostro software. MPLAB offre la possibilità di conta-



```
programma esempio;
var
base:integer;

procedure  somma(uno, due:integer);

        var
        sol:integer;

        begin
        sol=uno+due;
        writeln ('La somma è :',sol);
        end;

begin
clrscr;
base :=1;
somma(14,12);
somma(25,base);
end.
```

Esempio di programma con chiamata a procedimento in un linguaggio di alto livello.

```
Dichiarazione variabili
INTERO: N, somma
fine dichiarazione variabili

inizio

da N=1 fino a N=200 fare
  Somma <- Somma_N_Naturali(N);
  mostra sul display ('La somma dei ',N,'
  primi naturali è ','Somma)
fine da

fine
```

```
funzione Somma_N_Naturali(INTERO N) :
INTERO
variabili
INTERO: Somma,i

Somma <- 0
da i<-1 fino a i=N fare
  Somma <- Somma+i

Risultato <- Somma
fine funzione
```

Dichiarazione di una funzione.

```
funzione NOME (arg1,...,argN) : TIPO
variabili
..... {si dichiarano}

        azione 1
        .....
        azione N
        Risultato <- Valore
        Fine funzione
```

Esempio di programma che utilizza una funzione.

si trasmette un indirizzo e, nel caso si verifichino delle variazioni nel contenuto dello stesso, queste saranno permanenti, anche una volta terminata l'esecuzione del procedimento.

Nell'esempio della figura vediamo che nella dichiarazione del procedimento si definiscono due variabili: uno e due, variabili intere. Il procedimento ha la struttura di un programma normale e realizza una o più funzioni (nel nostro caso somma due numeri e presenta il risultato sul display). Nel programma principale si può ripetere la chiamata al procedimento scrivendo il suo nome e trasferendo i parametri con cui desideriamo che lavori.

Funzioni

Le funzioni sono molto simili ai procedimenti, con la differenza che restituiscono un valore al termine dell'esecuzione. Il tipo di questo valore si dichiara al momento di dichiarare la funzione. Quindi, una funzione si definirà nel modo indicato dalla figura.

TIPO è il tipo di dato che restituirà la funzione, NOME è il nome dato alla funzione e arg1...argN sono i parametri che passeremo alla funzione. Al termine di tutte le azioni la funzione restituirà un risultato che sarà definito quando chiameremo la funzione.

Per chiamare una funzione dovremo aver definito nelle nostre dichiarazioni di variabili una variabile dello STESSO tipo di quello restituito dalla funzione. Quindi, dovremo assegnare a questa variabile ciò che ci restituisce la funzione nel seguente modo: Variabile <- Nome_Funzione (arg1,...,argN).

La struttura commentata, che si può vedere nell'esempio della figura, è quella corrispondente ai linguaggi di alto livello, e si semplifi-

